



Ivan Iwannado



Ivan Iwannado knows what he wants to do, but like most of us he'll take a while to figure out how to do it. Learn from his mistakes

THIS MONTH:

IVAN IMPROVES HIS DIGITAL PICS

Ivan Iwannado got a shiny new digital camera for Christmas, but he's not very good at taking pictures with it. Sometimes the software on his PC improves the images; other times, he can't get them to look any better. What can he do to improve his pictures?

While image-editing software can overcome some issues, as Ivan has discovered, some problems are easier to fix than others. If Ivan remembers a few simple rules, he'll get much better images.

So what can be fixed and what can't?

Let's start with exposure problems. If a scene is evenly lit, Ivan's camera won't have any problem taking a correctly exposed photo. If lighting varies, though, he may find bright regions are overexposed, and details such as tracks in snow are 'blown out' as featureless whiteness. Similarly, details in the shadows of an otherwise bright scene may be lost due to underexposure. These problems cannot be corrected in an image editor because the detail isn't captured so it can't be enhanced.

Before taking a photo Ivan needs to decide which is the most important part of the scene and use the camera's metering to ensure this bit is correctly exposed. Most cameras have a spot metering mode – you aim the camera straight at the key area of your scene, press the shutter button halfway down, then frame the photo as you want before depressing the button fully.

What if both dark and light areas are important?

On sunny days you may struggle to get both sunlit and shaded areas looking good whatever you do with the exposure settings or metering. Some cameras have a contrast setting that can be turned down to help the camera cope with such tricky lighting. Another good idea if Ivan's taking photographs of nearby objects in direct sunshine is to use flash to 'fill in' the shaded areas. Most cameras have a 'flash always on' option. This will make the shadows less dark, so the camera will produce a more evenly exposed image.

Ivan should also remember he can look at the picture as soon as he's taken it; this is one of the main benefits of digital cameras. If the first picture looks too bright, he can try lowering the exposure.

One thing that drives Ivan crazy is when he takes photographs in large buildings or at night using the flash. He ends up with really dark pictures – what's the solution?

The problem is that the flash units on most cameras will light a region up to around three metres from the camera. Even a dedicated flash unit will only extend this to around five metres. While image-editing software can manage some correction to overexposure, it's useless with underexposure. If the picture is pitch-black, that's the way it will stay.

So what can Ivan do?

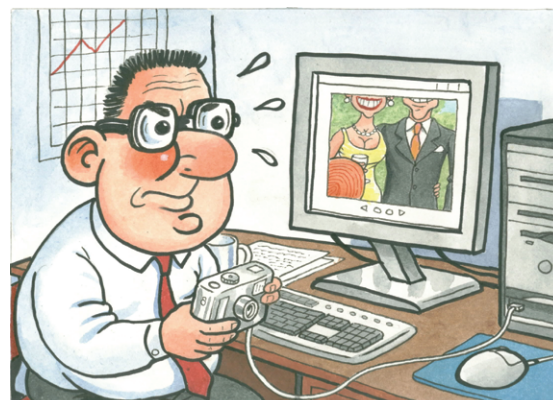
If Ivan's determined to take a photograph in a large gloomy building, altering the ISO setting on the camera to 400 or even 800 can increase the distance at which the flash will work. The only snag with this is that at high ISO settings, the image will probably look grainy due to digital noise. Alternatively, Ivan could get a tripod, turn off the flash and using a long exposure time – but he still won't get perfect results.

OK. Another problem Ivan has noticed is that when he wants to print a really big version of his best pictures, the quality isn't good enough.

Most digital cameras offer a default setting that's suitable for normal-sized prints – around 6x4". This has the advantage that you get a lot of photographs on your memory card, but does mean the images aren't detailed enough for larger prints. If Ivan wants to print larger images, he needs to choose one of the higher image quality options. If Ivan's camera allows it, he should also choose a larger image file size. The larger the file, the less compression is used on it, so the higher the quality of the image.

Yes, but then Ivan can only put a few pictures on his memory card.

Then buy a bigger card – they are not that expensive. In *Shopper* March 2006, we did a Labs review of 18 512MB memory cards, starting at just £19 including VAT. This size card should mean Ivan no longer has to choose between good images and lots of images. If Ivan has lots of storage space, he can also use the old photographer's technique of 'bracketing'. This means you take the same photo at three exposures: the one you think you want, and one either side. Many cameras can be set to do this automatically.



The only thing Ivan needs to be careful about is resizing his images before putting them online or emailing them to his friends. People won't want to get a 10MB picture of Ivan's goldfish sent to their dial-up network connection.

How big a file does Ivan need for his large prints?

The critical consideration is dots per inch, or dpi. While 150 dpi gets you a reasonable-quality print, you need 200dpi to 300dpi for a really sharp photograph. If you multiply the dpi by the size of the photograph you want, you'll get the size in megapixels. So a 10x8" image at 200dpi would need 10x200 by 8x200, or 2,000 by 1,600 pixels, which is 3.2 million pixels.

So if Ivan has a 3.5-megapixel camera, that will be fine for a 10x8" image?

Just about, provided the photo itself is perfect. If it's out of focus or there's motion blur, due to the subject or Ivan's hand moving as the photo was taken, it will be obvious in a 10x8" enlargement. The skill of the photographer is at least as important as the specification of the camera.

Fair enough. One thing that confuses Ivan is that even when he's got a photograph that looks good on the PC screen, it doesn't necessarily print well.

That may be down to the printer or the paper Ivan's using. He needs to make sure he's using photo-quality paper and that the printer is configured to use the high-quality paper.

Ivan is a happy man. He's got some great rules for getting better photographs; now he's going down to the garden to get a good picture of Jaws, his goldfish.

Nigel Knowitall

If you want to get something done using software but are unsure how to do it, fear not — help is at hand from our resident expert, Nigel Knowitall



FIXING TEXT FLOW AROUND IMAGES

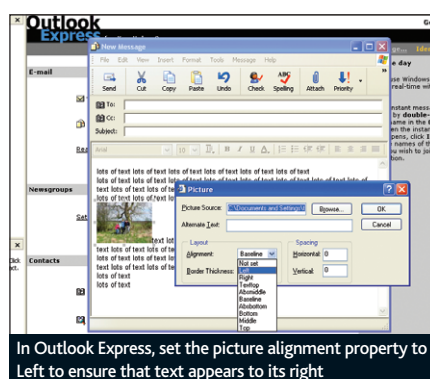
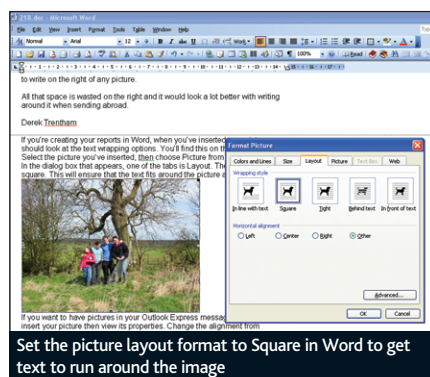
Q I write a lot of reports in Word with embedded pictures and also send them by email using Outlook Express. When I insert a picture in a document, it pushes the text that comes after the picture down the page. This creates a lot of wasted space to the right of the picture.

Can you tell me how to make the text flow around a picture, in the same way as words run around a picture in *Shopper*, for instance? This would save space and look a lot better.

Derek Trentham

A After you've inserted your picture you need to adjust the text-wrapping options, which can be reached via the Format menu. Select the picture you've inserted, then choose Picture from the Format menu. In the dialog box that appears, select the Layout tab. Here you'll find a choice of wrapping styles, and the one you want is Square. This will ensure the text runs around the picture as you want.

You can do a similar thing with pictures inserted in to Outlook Express messages. After inserting your picture, view its properties. Change the alignment from Baseline to Left, and you should find the text fills the area to the right of the image.



MATCHING DATA BETWEEN SPREADSHEETS

Q I work in a secondary school and routinely copy student data from one spreadsheet to another. The first spreadsheet contains the academic results of a group of students. The surname, forename and date of birth are in columns A, B and C:

Student	Forename	Date of birth
JONES	Ian	18/08/1993
SMITH	Anne	06/12/1992
JONES	Peter	20/02/1993

Their gender and marks for various subjects are in columns D, E, F and G. The second spreadsheet has the same column headings, and columns A to C already contain the names of the students whose marks I want to transfer from the first spreadsheet. The names in each spreadsheet are mostly the same, but vary a little as new students arrive and others leave.

I spend hours checking and inputting data, and I'm sure this job can be automated. I am fairly used to Excel, but I'm not familiar with writing the code for macros. Can you help?

Manjit Sahota

A The first tricky part of this problem is locating matching students in each spreadsheet. As you can see from the example above, the surname alone won't uniquely identify students. Even a surname and forename can be duplicated, so to identify a student, our macro has to check the surname, forename and date of birth combined. In an ideal world, each student would have a unique student number or ID. We can create something similar in a new column A of both spreadsheets. This column is set to contain the surname, forename and date of birth concatenated into a single string using the formula =B2&C2&D2. This is copied down the column using the fill handle so the row number changes automatically.

The next decision is how to work through the data. Do you want to visit every row in the source spreadsheet and look for matches in the destination spreadsheet, or take each row in the destination spreadsheet and see whether there's data in the source to copy over? It seems more natural to start with the source data, but it is actually more efficient to work through the destination spreadsheet. That way, you'll only spend time trying to find matching data for rows you need to complete.

You also need to work out what happens if you find a match, but there's already data in the destination cell. We've chosen to overwrite the data with the value from the source data, but you could easily have a different rule. Finally, you need to

work out what to do if the cells in the source sheet are empty but there are values in the destination sheet. We've chosen to leave the destination data unchanged if the matching source row is empty.

The full macro is shown below. It assumes that the source data is stored in a spreadsheet called sourcedata.xls and the destination data is stored in destdata.xls. Each spreadsheet is made active, then the UsedRange property is invoked to find the block of cells containing data in each. Next the macro begins looping through each row in the destination spreadsheet using:

```
For Each rw In mydest.Rows
```

For each row it first looks for a student with the same surname, forename and date of birth in mysource. To do so, it uses the .Find method. It stores the concatenated surname, forename and date of birth of the current student in thisstud and then looks for a matching value in mysource using:

```
With mysource
Set c = .Find(thisstud, LookIn:=xlValues)
```

If anything is found, c contains the reference to the cell where it is found. Otherwise, c is empty.

The code that checks whether the source record found contains data or not is:

```
If mysource.Cells(c.Row, 5).Value <> "" Then
    rw.Cells(1, 5).Value = mysource.
        Cells(c.Row, 5).Value
End If
```

If you wanted to overwrite the destination data, no matter what the source data contained, you should remove the If..Then line and the End If line, so the code would simply read:

```
rw.Cells(1, 5).Value = mysource. Cells
    (c.Row, 5).Value
```

You'd do the same thing every time there was a similar check on the value in mysource.cells(c.row)

Here is the full macro. If you had a lot of cells to copy over, you could create a loop in place of all the If..Then statements:

```
Sub Macro1()
Windows("sourcedata.xls").Activate
Set mysource = ActiveSheet.UsedRange
Windows("destdata.xls").Activate
Set mydest = ActiveSheet.UsedRange

For Each rw In mydest.Rows
```



```

thisstud = rw.Cells(1, 1).Value
With mysource
Set c = .Find(thisstud, LookIn:=xlValues)
If Not c Is Nothing Then
    If mysource.Cells(c.Row, 5).Value <> "" Then
        rw.Cells(1, 5).Value = mysource.Cells(c.Row, 5).Value
    End If
    If mysource.Cells(c.Row, 6).Value <> "" Then
        rw.Cells(1, 6).Value = mysource.Cells(c.Row, 6).Value
    End If
    If mysource.Cells(c.Row, 7).Value <> "" Then
        rw.Cells(1, 7).Value = mysource.Cells(c.Row, 7).Value
    End If
    If mysource.Cells(c.Row, 8).Value <> "" Then
        rw.Cells(1, 8).Value = mysource.Cells(c.Row, 8).Value
    End If
End With
Next rw
End Sub

```

EMBEDDED HYPERLINKS WITH EXCEL

Q I have a question regarding Excel and embedded hyperlinks. I have a spreadsheet and wish to link a cell to an image of an invoice. I have got this working fine on my PC, but next I want to copy the Excel file and a directory of images to a CD so I can give it to my accountant who can then view the invoices from the spreadsheet. Excel hyperlinks appear to store an absolute path to image files, such as K:\Data\MyCompany\Images\0003.JPG. It does this even if I refer to them in the Edit Hyperlink window (Ctrl-K) using a relative path such as ../Images/0003.JPG.

When I transfer the pictures in K:\Data\MyCompany*.* to the CD, so they reside in E:\MyCompany\, Excel still refers to K:\Data\MyCompany\ and all the links are broken. I can save the Excel file as XML and edit the links, but is there a neater method within Excel?

Roger

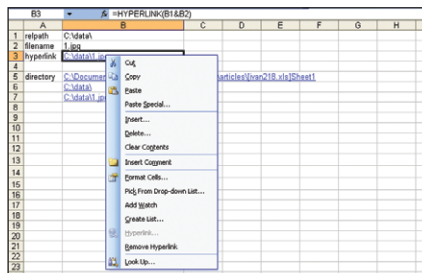
A The solution to your problem is to use the Hyperlink worksheet function rather than the Hyperlink editing dialog. You can create a hyperlink by entering a value such as:

```
=Hyperlink("C:\data\3.jpg")
```

This will do the same thing as what you've already got. But the Hyperlink function is much more flexible, and its arguments can be formulae themselves. For instance, you could store the folder name for your files in cell B1 and the filenames in cells B2, C2, D2 and so on. Then you could create hyperlinks by entering in B3 the formula:

```
=Hyperlink($B1&B2)
```

Copying this formula across the row, you'd get a series of Hyperlinks like this:



Use the Hyperlink worksheet function to create hyperlinks that can be changed easily

	A	B	C
1	Folder	C:\data\	
2	Filename	1.jpg	2.jpg
3	Hyperlink	=Hyperlink(\$B1&B2)	=Hyperlink(\$B1&C2)

Now if you move the files to a different folder, you need only update cell B1 to restore the links.

There's an even better solution than this, though. The Excel function Cell can be used to find out a variety of information about the current workbook and operating environment. If you use:

```
=Cell("filename")
```

the function will return the full filename and path in which the current workbook is stored, including the drive letter. If our spreadsheet is stored in C:\Data\ and is called Ivan218, the function Cell("filename") will return:

```
C:\Data\[Ivan218.xls]Sheet1
```

There are various ways you can use the Cell function to create a self-updating hyperlink address. You could pull the folder name from the result of the Cell("filename") function, using the Left function to get the main folder. To do this you need to look for the \ symbol that occurs closest to the end of the filename, then take all the characters to the left of that.

This is quite a tricky thing to achieve, but as finding the right-most occurrence of a character within a string is useful in a variety of circumstances we'll use this method. First you need to find out how many occurrences of the character there are in the string. To do this, you find the length of the text if you replace all the occurrences of the text you're looking for with a null string. Subtracting this from the original length gives the number of occurrences. If the Cell("filename") function is in B5, you'd start with:

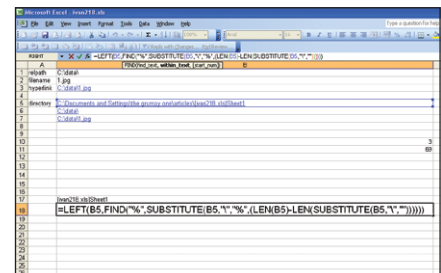
```
Len(B5) - Len(Substitute(B5, "\", ""))
```

Substitute expects at least three parameters: the text to work on, the value to be replaced and the value you want to replace it with. So Substitute(A1,"ivan","nigel") will replace all occurrences of the text ivan in cell A1 with the text nigel. In our case, we're replacing the \ character with nothing.

Then use Substitute again, along with the Find function, to get the final occurrence of the text you're looking for:

```
Find("%", Substitute(B5, "\", "%"), (Len(B5) - Len(Substitute(B5, "\", ""))))
```

You should recognise the latter half of this formula as the formula for the number of occurrences of \.



Use the Find and Substitute functions to extract a folder from a cell value containing the filename and path

But what does the rest of it mean? Let's start with the portion that reads:

```
Substitute(B5, "\", "%", (Len(B5) - Len(Substitute(B5, "\", ""))))
```

This uses Substitute as before. The important change is the use of a fourth parameter. Substitute can accept an instance number that specifies which occurrence of the old text you want to replace with new text. If you specify an instance number, only that instance of old text is replaced. So if there are five occurrences of \ the formula replaces the fifth instance only with %. However many occurrences of \ there are, it always replaces the last one.

Consider our initial filename and path of:

```
C:\Data\[Ivan218.xls]Sheet1
```

This will be replaced by:

```
C:\Data%[Ivan218.xls]Sheet1
```

The last bit of the formula uses the Find function to look for the % sign in the text. This assumes your filename and directory structure don't contain any % symbols; if they do, you'll need to choose a different character to mark the final \. The Find returns the position of the final \, which is the end of the file path. The final task is to use that information to pull out the path for use in your hyperlink:

```
=Left(B5, Find("%", Substitute(B5, "\", "%", (Len(B5) - Len(Substitute(B5, "\", ""))))))
```

You can use this as the path for all your images. As long as you keep the images in the same folder as the spreadsheet, the links will always be correct.

If your images are only ever in one of two locations – either a specific folder on your PC or a specific folder on a CD – there's a simpler solution. You can use an If statement to look at the drive letter of the current location of the spreadsheet, and work out the correct path for the hyperlinks from that. For example:

```
=If(Left(Cell("filename"), 1) = "C", "C:\data\", "E:\dif\data\")
```

NEXT MONTH

SOFTWARE HELP

Next month our new *Software Help* section will deal with all queries relating to office and creative applications. Send your questions to the address below

softwarehelp@computershopper.co.uk